A Wide-Spectrum Approach to Modelling and Analysis of Organisation for Machine-assisted Decision-Making

Vinay Kulkarni[1], Souvik Barat[1], Tony Clark[2] and Balbir Barn[2]

[1] Tata Consultancy Services, 54B, Industrial Estate, Hadapsar, Pune, India
[2] Middlesex University, London, The Burroughs, Hendon, London, NW4 4BT
{vinay.vkulkarni, souvik.barat}@tcs.com and {T.N.Clark, B.Barn}@mdx.ac.uk

Abstract. This paper describes a modeling approach that helps to represent necessary aspects of complex socio-technical systems, such as organization, in an integrated form and provides a simulation technique for analyzing these organisations. An actor-based language is introduced and compared to a conventional simulation approach (Stock-and-Flow) by simulating aspects of a software services company.

1 Introduction

Modern enterprises are complex systems involving multiple stakeholders that need to respond to a variety of changes within a highly constrained setting. The cost of an erroneous response is prohibitively high and may possibly reduce options for subsequent changes in direction. Large enterprises adopt an organisational structure best suited for ease of management and control; this can lead to undesirable side-effects such as scattered and fractured knowledge about goals, operational processes, IT systems, design rationale, IT infrastructure and best practices. Analysis of such an organisation relies almost exclusively on human experts who are expected to keep track of current and historical phenomena or traces (possibly using large collection of spreadsheets), and interpret them with respect to the problem under consideration; this is a huge challenge considering the size and complexity of modern enterprises [21].

Current practice advocates the use of multi-dimensional view-decomposition techniques to address complexity and size of an organization [1-7, 10, 16]. For instance, the Zachman Framework [1] recommends a two dimensional classification matrix describing six interrogative aspects that include *what*, *where*, *when*, *why*, *who* and *how* with five levels of perspectives (*i.e.*, *Scope*, *Business Model*, *System Model*, *Technology Model* and *Detailed Representation*) to represent an enterprise as set of interrelated views. This structured representation helps to improve documentation quality and visual navigability of enterprise artefacts to a large extent. However, the analysis support for one or limited views(s) results into view integration issue in practice [18,19]. The existing machineries for organizational decision making are mostly limited to one specific aspect only. For example, an organizational decision making process that involves goal, strategy, and business processes of an organization can only be addressed by integrating i*[1] (for goal aspects), iThink[2] (for strategic aspects) and BPMN tools[3] judiciously. The paradigmatically diverse nature of languages for goal modelling, Stock-and-Flow (SnF) modelling, and business process modelling means it is difficult to come up with a single meta language in terms of which the three modeling languages can be expressed. As a result, one is left with no recourse but to establish suitable mapping relationships between the relevant model elements across different models for co-simulation [25]. Clearly, such a method has to rely heavily on human expertise for correctness of execution. Moreover, being at model-level and models being purposive, the mapping relationships are rarely reusable across different contexts let alone different problems thus making the modelling endeavour both effort and cost intensive. All in all, current state of practice suffers from several limitations as regards comprehensive and efficient support for organizational decsion making.

Our claim is that the problem of organizational analysis and support for decision making that involves multiple aspects, such as goal, strategy and operational aspects, can be addressed by simulation using a single language. Importantly, a single language for representing an organization in the context of decision making solves the view integration issue; and simulation can be used to support various forms of what-if and if-what analysis for organizational decision making.

This paper introduces Enterprise Simulation Language (ESL) for organisational analysis and simulation. To validate our claim that ESL based approach is more appropriate than existing analysis and simulation machineries that support a single organizational view (or subset of views) we define a simple real-world case study and represent it using both, a representative leading current approach Stock `n` Flow (SnF) and ESL. We carried out analyses that predict likely impact of modifying control variables onto the observable variables of the system at steady state. We show that the ESL approach is at least as good as SnF in terms of producing analytical results for operational strategy in organizational decision making, and that the SnF model exhibits problems that are not present in the ESL specification.

2 State of the Art

Enterprise decision making is a complex process and it requires precise understanding of enterprise in terms of goals, strategies, operational process, organizational structures, constraints, etc., and ability to analyse and predict them in dynamic environment. Traditional domains like engineering, manufacturing, military, traffic and control industry use computer aided decision making to reduce the human dependency and increase precision. Several projects those are based on CIMOSA[2], TOVE[3], PERA[4], GERAM[5], GRAI[6], DoDAF[7], MoDAF[4] have benefited significantly by adopting methods that supports quantitative and qualitative simula-

[1] http://www.cs.toronto.edu/km/istar
[2] http://www.iseesystems.com/Softwares/Business/ithinkSoftware.aspx
[3] http://www.softwareag.com/corporate/products/aris/default.asp
[4] https://www.gov.uk/mod-architecture-framework

tion. While simulation has evolved into a mature discipline in traditional domains, its application to complex socio-technical systems such as organisations is relatively new [8,9]. The challenges in this domain are, in part, posed by the informality of the specification and dominance of symbolic and qualitative approaches in practice.

The current state-of-the-art of enterprise specifications in information system are broadly classified into two: those focusing on operational aspect that includes *what*, *how* and *who*, [10, 11] and those focusing more on the goal aspect that includes *i.e.*, *why* and *who* [12, 13 and 14]. Supporting machinery for the former is best seen as a means to create high level descriptions that are meant for human experts to interpret in the light of synthesis of their past experience. The stock-n-flow model [15] provides an altogether different paradigm for modeling high-level *operational strategy* (the *what* plus abstract specification of *how*) and comes with a rich simulation machinery (iThink) for quantitative analysis using averaging out models and formulae. Several BPMN tools, such as ARIS and Bizagi[5], providing simulation capability exist but they are limited to the *how* and *who* aspects only. Supporting infrastructure for goal aspects is comparatively more advanced in terms of automated deduction. However, they are largely removed from operational aspects. There are limited specifications, such as Archimate [17] and EEML[16], that balances goal and operational aspects but they are lacking in precise execution/simulation semantics. Hence they solve the integration and interoperability issues [18,19] but are not capable of promoting machine assisted techniques, such as simulation, in decision making.

At present, the only recourse available is the use of a method to construct a tool-chain of the relevant set of analysis and simulation tools with the objective of answering the desired questions. The projects based on TOGAF and Zachman adopt a standardized method but the use of analysis and simulation tools is still a challenge.

3 Our Approach

An enterprise can be understood well by understanding *what* an enterprise is, *how* it operates and *why* it is so. It further provides clarity on organizational responsibilities by understanding the *who* (*i.e.*, responsible stakeholders for *what*, *how*, and *why*) aspect of the organization. We argue that the information about *what*, *how* and *why*, augmented with *who*, aspects of an enterprise is necessary and sufficient for organizational decision making activities under various operating conditions. The further proposition of our work is the need for a language to specify these aspects as a single integrated model. A single unified specification is believed to be a viable option to overcome present problems due to multiple partial views that need to be integrated for deriving insights obtainable from their analysis and simulation. An executable model can enable what-if and if-what scenario playing thus leading to a priori, data-driven, and informed decision making. Naturally, such a specification language needs to be close to execution infrastructure and therefore quite a distance away from the end-

[5] www.bizagi.com

user decision-maker. This abstraction gap is best bridged by developing domain specific language(s) (DSLs) each describing the *why*, *what* and *how* aspects for all relevant stakeholders or the *who* aspect in a localized relatable 'business facing' form specific for an analysis use-case. As both languages express the same aspects albeit at different levels of abstraction, it is possible to automatically transform a DSL specification to the simulatable closer-to-machine language (ESL). We propose model based engineering to address these practical issues. Fig 1 presents a pictorial description of this approach.

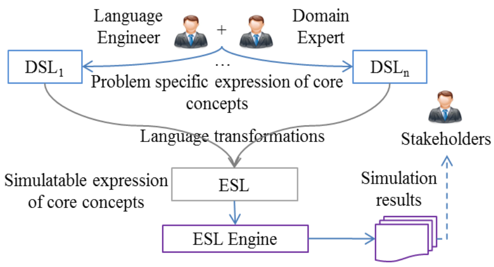


Fig. 1: High Level Approach for Organizational Decision Making

4 ESL

From an external stakeholder perspective, an organisation can be viewed as something that raises and responds to a set of events as it goes about achieving its stated goals. Organisations consist of many autonomous units, organised into dynamically changing hierarchical groups, operating concurrently, and managing goals that affect their behaviour. We describe structure and behaviour of an organization using a small set of concepts and their relationships as depicted in Fig 2. An *Abstraction Unit* is the core conceptual element that can represent organization unit, people as well as IT systems. It interacts with the environment through *inEvents* and *outEvents*. It exposes a set of *Goals* as its intent, *Levers* as possible configuration parameters and *Measures* that describe qualitative and/or quantitative evaluation of its state to the external stakeholders. An *Abstraction Unit* is defined in terms of *Data* as a set of state variables/state variables, *History* as a set of past states, *Behaviour* as what the unit "does", and sub-units that it may compose of. Our hypothesis is that these concepts are necessary and sufficient for specifying *why*, *what*, *how* aspects from the perspectives of the relevant stakeholders (*i.e.*, the *who* aspect) of an organization. The structural elements *Abstraction Unit*, *In Event*, *Out Event*, *Data* and nesting capability of *Abstraction Unit* specify the *What* aspect, *Goal* specifies the *Why* aspect, *Behaviour* specifies the *how* aspect and *Abstraction Unit*, as individual, defines the *who* aspect of an organization.

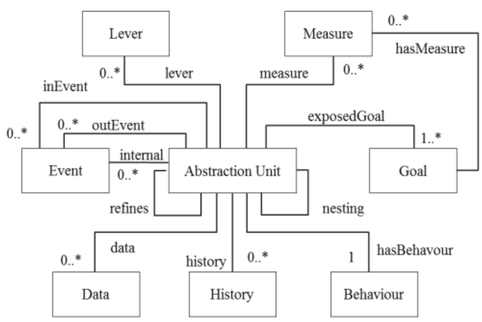


Fig 2: High Level Conceptual Meta Model

In addition, we argue that *Abstraction Unit* must support a set of characteristics of organization theory [20] as essential requirements for modeling basic elements of organizations, *i.e.*, organizational units, human elements and IT system components [21]. These charateristics are as follows: **reactive** *i.e.*, Abstraction Unit must respond appropriately to its Environment; **adaptable** *i.e.*, Abstraction Unit may construct and reconstruct its structure in its life cycle [20]; **modular** *i.e.,* Abstraction Unit must encapsulate structure and behaviour of organisation; **autonomous** *i.e.,* Abstraction Unit is a standalone entity; **intentional** *i.e.,* Abstraction Unit has its own goals; and **compositional** *i.e.*, an Abstraction Unit can be an assembly of Abstraction Unit. The *In Event* and *Out Event* concepts help to capture reactive nature of Abstraction Unit, the intent is captured using *Goal*, modularity is achieved through *Abstraction Unit*, autonomy is possible due to the concept of *Internal Event*, and composition can be specified using nesting relation. We use these concepts as the basis for ESL language definition where adaptability is incoporated using the operational semantics of ESL.

We aim for the ESL to reflect these core concepts by having an operational semantics based on the Actor Model of Computation (AMC) [22] and its relation to organisations, *i.e*., iOrgs [23]. Computationally, an Actor has an address and manages an internal state that is private and cannot be shared with other actors in the system. Execution proceeds by sending asynchronous messages from a source actor to the address of a target actor. Each message is handled in a separate execution thread associated with the target of the message and the message itself (collectively referred to as a task). During task-execution an actor may choose to change its state and behaviour (becoming a new actor) that is immediately available to process the next message sent to the target address. AMC is also appropriate to support the key characteristics listed such as:- **reactive**: Actor interacts with other Actors; **adaptability**: An actor changes behaviour as a result of handling each message; **modularity**: Access to an actor's hidden implementation is provided by a message interface; **automomy**: The AMC is highly concurrent with each actor being able to spawn multiple threads and over which other actors have no control; **intent**: Actors can form the basis for Multi-Agent Systems [24]; **composition**: Actors can be nested.

We are implementing a robust execution engine for ESL as part of a larger initiative for developing pragmatic framework for organizational decision making[6]. A prototype of ESL execution engine has been implemented in the programming language Racket[7] and all results in this paper are generated by our implementation.

4.1 Decsion Making using DSL

ESL supports decision making through *what-if* (what would be the consequence in terms of *Measures* if *Levers* and *Organization Unit* definition are specified) and *if-what* (what *Levers* and *Organization Unit* would have led to a consequence) scenario playing. Organization under consideration can be modelled as an instance of the meta model in Fig 2, behaviour and conditions describing the impacts of goals in a behaviour can be encoded using ESL. In a *what-if* simulation, if the *Measures* are not within

[6] www.tcs.com/about/research/research_areas/software/Pages/Model-Driven-Organization.aspx
[7] http://docs.racket-lang.org/drracket/

expected range then error could either be with the model (*i.e.*, organization definition) or with values supplied to the *Levers*. The *if-what* simulation requires sophisticated guidance to arrive at meaningful organizational structure and values to the *levers*. Current ESL execution engine implementation is capable of *what-if* simulation.

5 Case Study

In this section, we consider a software provisioning organization to illustrate simulation of organizational aspects in the context of organizational decision making. For space reason, we limit our discussion to operational strategy or the *what* aspect in this paper. The organization under consideration bids for software development projects in response to requests for proposals (RFPs) based on customer profile and price considerations. Once acquired, a project is resourced and executed using tried-and-tested development processes, finally delivering to the customer and releasing resources. This business as usual (BAU) scenario involves operational strategies including skill-matching, dealing with unforeseen demand, staff attrition, resource utilisation, accounting for operational delays, while ensuring business targets are met.

In addition to maintaining its BAU state, the organisation needs to decide upon strategies to improve its BAU state. An increase in similar projects should improve the maturity of the workforce with consequent improvements in productivity, quality and track-record. Project costs can be reduced by employing fewer resources or less experienced resources, but this might increase the pressure on the workforce inducing undesirable delays. Investment in training and productivity tools might mitigate the delays but incur their own costs and delays, however this might be balanced by the

Demand Characteristics	Simulation Value
Project Arrival Rate(count/month) of <LMLR, MMLR, MMHR, HMHR>	<10, 8, 8,6>
Project Size of LMLR, MMLR, MMHR, HMHR (Range in KLOC)	<10-40, 10-40, 40-100, 40-100>
Chargeable Rate of <LMLR, MMLR, MMHR, HMHR> (K USD / KLOC)	<100,200,250,400>

MMLR (Medium margin and low risk)	HMHR (High margin and high risk)
LMLR (Low margin and low risk)	MMHR (Medium margin and high risk)

Bid winning rate <40, 30, 30, 30>

Supply Characteristics	Simulation Value
Resource Availability of <J, SJ, S, E> (count)	<80,80,70,60>
Compensation of <J, SJ, S, E> (K USD/month)	<7,8,12,15>
Attrition Rate of J, SJ, S, E (%)	<10,10,10,10>

Skilled Junior	Expert
Junior	Senior

Joining probability <70, 70, 60, 60>

Software Provisioning Organization

Levers that can be tweaked to alter organizational characteristics	Simulation Value
Required Effort	As per COCOMO equation
Bidding Cost of <LMLR, MMLR, MMHR, HMHR> (K USD/Project)	<20,30,30,40>
Setup Cost of <LMLR, MMLR, MMHR, HMHR> (K USD)	<100,200,100,200>
Productivity of J, SJ, S, E (as multiple of standard productivity as per COCOMO)	<0.9,1,1,1.1>
Resource Distribution of J, SJ, S, E	<25,35,20,20>
Initial Resource Strength and Bench Strength	<1600,750>
Productivity Tool Cost < Initial Cost (K USD), Recurring Cost K USD/Year, Tool Training Cost/ Usage>	<1000, 250/Year, 20/Project>

Receive RFPs → Submit response → Bid win → Execute project → Deliver

Expense, Revenue, On Time Service Delivery

Fig 3: Case Study Description

future profit margins that are supported by better quality processes. The decision to adopt a tool-centric approach hinges upon being able to compute return on investment in tools and the break-even point at which productivity and quality gains start to outweigh costs.

5.1 Description and Implementation

Fig. 3 shows such a software provisioning organization, which operate in an environment that comprises of a *Demand* and *Supply*. In this case study, the Demand is characterized by four kinds of software development projects: low margin low risk (LMLR), medium margin low risk (MMLR), medium margin high risk (MMHR) and high margin high risk (HMHR). The characteristics of these projects, *i.e.*, typical project arrival rate, standard chargeable rate and size of the projects, are specified in a table that associated with Demand in Fig. 3. We consider *Resources* are the primary element that forms the Supply of this organization. The case study considers four kinds of workforce resources: junior (J), skilled junior (SJ), senior (S) and expert (E). Individual characteristics of resources are also depicted in Fig. 3. The software provisioning organization has a goal to improve profit margin without compromising service qualities (*i.e.*, on time delivery of the software with desired quality). The organization measures three metrics: revenue, expense and timely delivery, to keep track of the BAU state.

Internally, organization adopts different strategies, by defining best possible values to the levers and organizational structure, to fulfil desired goals. The internal levers of software provisioning organization are depicted in a table within organization in Fig 3. The *Bid Winning Rate* and *Joining Probability* of Fig 3 are also considered as levers of software provisioning organization but they are not completely controlled by the organization, rather they have influence on external factors as well. For example, the efficacies of internal sales strategies determine the initial *Bid Winning Rate*, but in a long run it is largely influenced by track record and market perception of the ability to deliver a project with increased service quality. The internal structure of the organization in terms of abstraction units is depicted in Fig 4. Each unit is implemented as an ESL actor. The Customer Unit and External Resources interact with Service Provisioning Organization Unit. The Service Provisioning Organization Unit is realized as four interacting Units – Sales, Delivery, Resource Management and Account Units. The bidding process and project execution processes are encoded within Sales

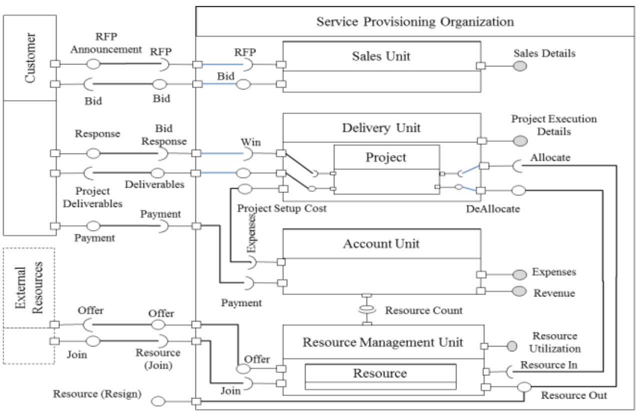


Fig 4: Organizational Structure of Software Provisioning Organization

Unit and Delivery Unit respectively. The recruitment strategy and current resources are managed by the Resource Management Unit. The resources are also modeled as an abstraction unit (or actor) where each resource managing its own productivity, leaving strategy, *etc.* and they move between Resource Management Unit and Project Unit based on allocation and deallocation requests. The Account Unit coordinates with all other internal units and updates the state of Revenue and Expenses. The organization builds up good track record by delivering projects on or before committed delivery time as well as meeting or exceeding deliverable SLAs, thereby influencing the bid winning rate.

In order to run a simulation we need an initial data model (the levers and other internal variables). For example, LMLR projects are charged at the rate of 100K USD/KLOC resulting in 40% awards with 10 projects arriving every month a standard delivery time. Junior skilled resource is paid 8k USD/month. At this pay package, 70% of the selected candidates join and 10% of existing skilled juniors leave. Recruits join in monthly batches with random resignations. The initial values of rest of the levers are depicted in Fig 3 as *Simulation Values* in the tables. We implemented a Racket based GUI for setting *Levers* values and observing *Measures* of a simulation run. Customized GUI interacts with generic ESL execution engine to supply input data and get Meassures values from a simulation run.

5.2 Simulation & Results

The organization is faced with several business-critical decisions such as: Are resources optimally loaded or is there some slack? Will quoting a reduced price or delivery time be more effective at winning more bids? Will staff training or the use of productivity tools reduce delivery time? When would the benefits start outweighing the costs? What J:SJ:S:E configuration delivers optimal KPIs? What would be the impact of scarcity of Expert resources on KPIs? What would be the result of focusing on high margin projects only?

We consider a goal of increased revenue without compromising the service quality. Management would like to have as precise answers as possible to the following kind of questions: Q1: Are we operating in a comfort zone? If so, how far can one go by removing existing slack (without compromising desired service quality)? Q2: How far can one go with existing workforce distribution (J:SJ:S:E)? Q3: For this organization setting, what is the best workforce distribution possible *i.e.,* a local optimal situation? Many other questions are associated with the decision making that may lead to global optimum solution. In this paper, we limit our discussion to first two questions due to the space limitation.

Simulation Result of Q1: Fig. 5 shows the *Measures* of a simulation run when *Lever* values are as depicted in Fig 3. The horizontal histograms depict the number of RFP Received, RFP Responded, RFP Won, Projects completed on time, Projects completed with delay and Project Pipeline (from to bottom to top), where four colors of each histogram (except project pipeline histogram) represent the metrics related to LMLR, MMLR, MMHR and HMHR respectively. The vertical histograms depict the Revenue, Expenditure, Profit and Saving due to productivity tool respectively (from left to right). The organization is winning about 33% of bids all of which are being executed within the expected time. Also, there is hardly any project that is not able to start due to non-availability of resources. Clearly, the organization seems to be operating in a comfort zone; how much more can the existing workforce deliver? The organization needs to win more bids. Delivery time and cost are the variables influencing bid winning percentage. We change the chargeable rates for LMLR, MMLR, MMHR and HMHR projects from {100, 200, 250, 400 (all in $K per KLOC)} to {90, 180, 225, 350} thus improving their bid winning percentage from {40, 30, 30, 30 (all in percentage)} to {90, 70, 70, 70}. Fig. 6.a shows the effect of this modification on measures. Bid winning percentage improves to 76% from 33%. The number of projects completed on time remains more or less the same but there is significant increase in the number of projects delivered late. Also, a significant number of projects witness delayed start due to non-availability of resources. Significant increase in bids won results in significantly high revenues even when chargeable price is reduced. With expenses remaining more or less the same (linked largely to number of resources on board) profits increase significantly.

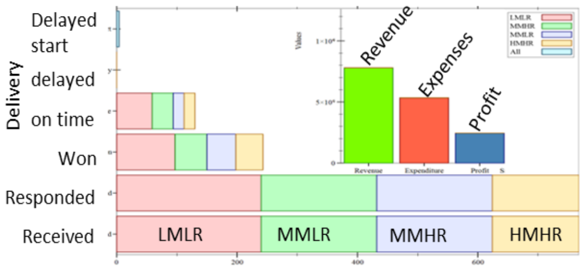
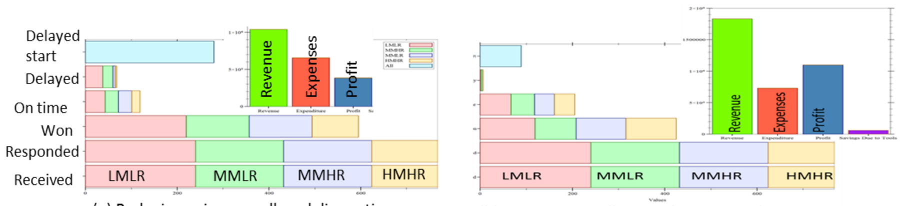


Fig. 5 Measures of a simulation run with initial values

Simulation Result of Q2: As seen from Fig 6.a, delayed delivery and project kick-off queue build-up are critical concerns. How can these concerns be effectively addressed keeping the resource distribution unchanged *i.e.*, J:SJ:S:E::25:35:20:20? Clearly there is a need to increase workforce productivity. One can think of having a better trained workforce or a better-tooled workforce or both. We change productivity of junior, skilled junior, senior and expert from {0.9, 1, 1, 1} (all as a factor of standard COCOMO[8] productivity metric) to {1, 1.1, 1.1, 1.1}. This comes at increased training cost for the four kinds of workforce from {10, 10, 10, 15} to {20, 20, 20, 25}. In this simplified model, we have not considered an adverse side effect namely, revenue loss incurred by the organization while part of its workforce was undergoing training. Productivity can be further increased by a factor of 1.25 by using tools. This too comes at tool license and training costs. Figure 6.b shows the effect of these changes



(a) Reducing price as well as delivery time

(b) Training as well as productivity tools

Fig. 6. Modifying Simulation Parameters

[8] Barry W Boehm. Software Engineering Economics. Prentice Hall, 1981 ISBN:0138221227

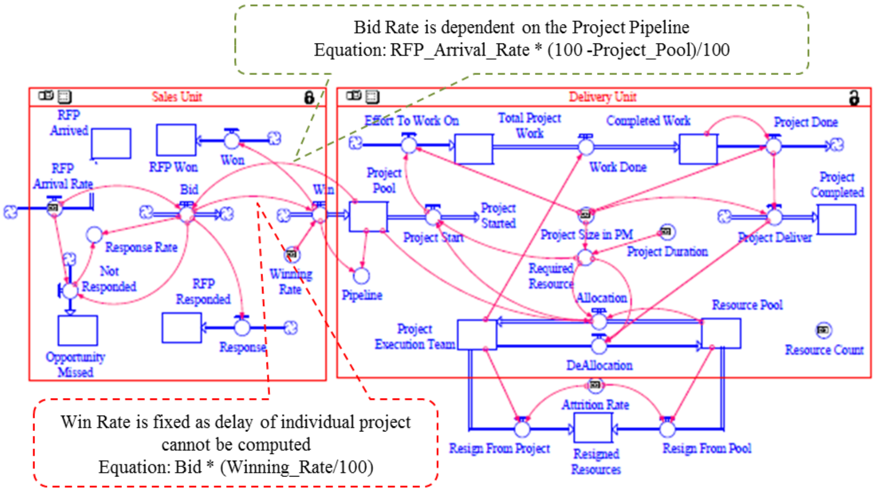


Fig. 7. Stock-and-flow model

on the measures. There is increase in the number of projects delivered on time. More significantly, no HMHR project is delivered late thus saving on delayed delivery penalties. Also, there is a significant increase in proportion of HMHR projects delivered on time. The sum total of all is a significant increase in revenue as well as profits. However, the delayed start for projects remains a cause for concern.

Simulation Results of Other Questions: Essentially, the management would like to have an idea of the possible outcomes (*Measures* – revenue, expenses, on time deliveries, resource utilization) for given set of operating parameters (*i.e.*, *Lever* values), and prediction on operating parameters (possible *Lever* values and organization structure) for a specific set of *Measures* as part of what-if and if-what analysis. We demonstrated two scenarios where *Measures* are observed by setting values to the *Levers* using what-if simulation runs. The rest of the analyses are not very different than above two. However, a complex decision making process involves an iterative scenario playing process that involves both kinds of simulations (*i.e.*, what-if and if-what) till a satisfactory result is observed. While reaching a satisfaction level an organization can be altered in many ways that include the change in lever values, reconstruction of organization structure (by adding new unit, refining existing units and/or both), refinement of sub-unit goals. Thus a method (simulation method) supported by ESL simulation engine is better suited for complex organizational decision making process.

6 Comparative Study

To compare the performance of the ESL simulation engine with existing simulation engines, we implemented the case study using Stock-and-Flow (SnF) model and simulated using iThink for the same set of what-if analysis. Fig. 7 shows a SnF model

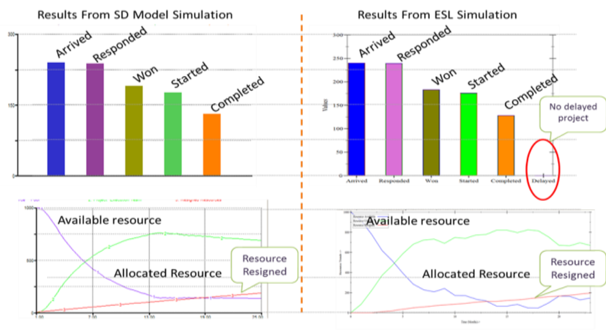


Fig. 8. Result comparison of SD Model and ESL Model with sufficient resources

that represents a generalized sub-set of the case study illustrated in earlier section (we generalize our case study as combinatorial and the decomposition of project and resource into the types lead to a 16-fold increase in SnF model size). The SnF model describes generalized behaviour of Sales Unit, Delivery Unit and Resource Management Unit where Sales Unit and Delivery Unit are represented without project specialization (LMLR, MMLR, MMHR and HMHR projects), and Resource Management Unit is represented without resource specialization (*i.e.,* junior, skilled junior, senior and expert resources). Lever values with ranges, *e.g*., *Project Size* and *Project Duration*, are simplified into fixed values to fit formulation using SnF model. The variable *Bid Rate* is dependent on the project pipeline, and *Bid Winning Rate* is dependent on the project delivery track record. Bid Rate is implemented using SnF equations but a fixed Bid Winning Rate is considered in SnF model as individual behaviour cannot be recorded in a generalized formation. For space reasons, we do not discuss Resource Management Unit, Account Unit and their interactions here. We specified an equivalent ESL model with Bid Winning Rate as dependent variable (instead of fixed value in SnF model) and observed the following results.

We simulated SnF and ESL models with input values <TotalResourceCount= 1000, AttritionRate=10, ProjectArrivalRate=10/month, ProjectSize=100PM, ProjectDuration=6 months, InitialWinningRate=80%> with the simulation results shown in Fig 8. The top row describes an overview (*i.e.*, number of RFP arrived, number of RFP responded, number of projects won, number of project started, and completed) and the bottom row describes resource trends (*i.e.*, free resource, allocated resources and number of resource resigned from organization). Simulation results are almost similar for SnF and ESL simulation engines. Steady state values of simulation results for resource trends are also similar. However, simulation output of ESL model is closer to reality with clear indication of resources joining a project in batches whereas resignations do not necessarily happen in chunks or specific interval.

We changed the operating environment by reducing available resources to half while keeping rest of the levers unchanged and observed results as shown in Fig 9. The

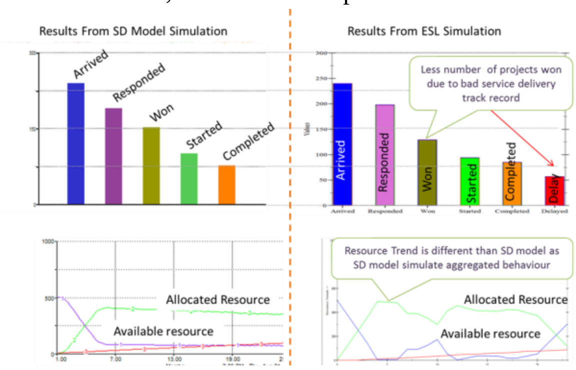


Fig. 9. Simulation run with Reduced Resources

SnF simulation can only say that, within the stipulated time period of simulation, how many projects got initiated and how many were completed. It is silent regarding project delays, whereas the ESL simulation clearly depicts the details of projects that got delayed and can provide additional important information regarding non-availability of resources and how they influence project delays.

7 Analysis

For the kind of decision making problem illustrated in this paper, current industry practice relies extensively on spreadsheets for data storage and arithmetic computation, and human experts for their interpretations. Such an approach typically represents the relationships between *Levers* and *Measures* in terms of static equations. The lack of support in expressing temporal aspects of an organization (including the interference between variables *i.e.*, *Levers* and *Measures*, with respect to time) limits the use of spreadsheets to being a data computation aid instead of data-driven decision making tool. Thus decision makers have to step in when it comes to predicting measures trends and the factors that influence various scenarios. For example, the number of projects won in a month/quarter/year for given arrival rate of LMLR, MMLR, MMHR and HMHR can be computed using spreadsheet; the number of projects completed in month/quarter/year can also be computed if number of resources are fixed; but predicting number of projects completed or project completed on time is not possible using spreadsheets when number of resources (*i.e.*, J, SJ, S, E) and their productivity changes with time and other factors.

SnF models are also used for this kind of decision making. In this approach, the system is specified in terms of stocks, flows of stocks, and equations over levers and variables that control the flows. The quantitative nature of SnF models and sophisticated simulation support enables decision making through what-if scenario playing. Expressing relationships between influencing factors or system variables over time is possible in such a dynamic model. However, it is best suited for an aggregated and generalized view of a system where individual details are abstracted out through averaging, and the sequences of events are grouped as continuous flows. This generalized approach and ignorance of individual characteristics that significantly influence the system over time often leads to a model that is somewhat removed from reality. For example, consider a policy that junior resource becomes a senior resource after working for 3 years. It is possible to capture this policy in a SnF model but the impact of improved productivity and additional cost for this junior-to-senior transition in a specific project cannot be detected in a model that uses averaging. Similarly, it is possible to determine the number of projects completed over time given a certain joining and resigning characteristics, but determining the number of projects delayed due to attrition is not possible. Since a resource is an individual actor in ESL and a set of individual actors participate in a project (instead of set average junior resources), it is possible to determine which specific project will be impacted. Similarly ESL can detect individual projects that are impacted due to resignations of allocated resources and junior-to-senior transition.

Though a SnF model is not intended for specialized behaviour, it is possible to argue that it can be specialized for such detailed analyses. But the effort required to specialize such models at the level of types leads to model size explosion *e.g.,* specialization of the notion of project into LMLR, MMLR, MMHR and HMHR projects and of the notion of resource into Junior, Skilled Junior, Senior and Expert resources leads to 16-fold increase in model size. Moreover, since SnF models offer poor support for modularity and change isolation, they are ineffective in dealing with industry-scale problems. For example, incorporating a minor change in the bidding strategy of a specific kind of project (*e.g.,* the decision not to bid HMHR projects when significant project pipeline is built up) may impact many flows and equations of SnF model.

We conducted a series of experiments to evaluate the desired characteristics of organizational decision making. We found that ESL scores better with respect to SnF in terms of modularity, compositionality, adaptability and intentionality. In different experiments, we also found ESL to be better than BPMN (and associated simulation engines) in terms of autonomicity, adaptability, intentionality; and better than intentional models, such as i*, in terms of autonomicity and compositionality. Moreover, ESL is capable of specifying the *why*, *what* and *how* aspects from the perspectives of a stakeholder (*i.e., who* aspect) in a localized and relatable form, which we believe is a significant advance if industry practice is to take to simulation as a decision making aid for socio-technical systems.

8 Conclusion

This research is being undertaken by research lab of organization in the business of offering software, processes and technology consultancy[9]. It is becoming increasingly apparent that coming up with the right solution and demonstrating its likely efficacy is lot more important and harder than implementing the solution. We believe formal modeling of relevant aspects of enterprise in a manner that supports what-if and if-what simulation holds the key. This paper has introduced an initial step to improve the use of simulation in organizational decision making. It is based on a working hypothesis that specification of the *why*, *what* and *how* aspects in a localized relatable form for all relevant stakeholders (*i.e.,* the *who*) is critical in overcoming limitations of the current state of practice. We have defined a simulatable language (ESL) for organisational analysis and support for decision making. The essence of simulation capability is demonstrated with a simplified real-world case study and compared simulation results with traditional simulation (SnF) on a prototype implementation of ESL engine. We conclude that our approach is highly promising for organisational analysis and that it offers advantages over existing approaches, but there is much work to do. Having added structure to simulation models we are in a position to introduce intentional behaviour in the form of both system-level and individual goals, using ideas from multi-agent systems. In addition, we will need to extend ESL with stochastic features in order to capture uncertainty in real life that leads to variability in

[9]www.tcs.com/about/research/research_areas/software/Pages/Model-Driven-Organization.aspx

the simulation. In longer term, we are working towards to a general purpose framework for organizational decision making where ESL based simulation engine will be used in coordination with business facing DSLs to enable business experts to pose their questions using business facing language and get their answers back in business terms. We think ideas from Fact Based Modelling (FBM)[26] could be useful especially as regards semantics.

9 Acknowledgements

We would like to thank Arun Bahulkar for his help in defining the case study.

10 References

1. John A Zachman. “A framework for information systems architecture.” IBM systems journal, 26(3):276-292, 1987.
2. Kosanke, Kurt. "CIMOSA—overview and status." Computers in industry 27.2 (1995): 101-109.
3. Fox, M.S., "The TOVE Project: Towards A Common-sense Model of the Enterprise", Enterprise Integration Laboratory Technical Report, 1992
4. Theodore J. Williams , Gary A. Rathwell, Hong Li., “A Handbook on Master planning and implementation for enterprise integration programs” , 2001
5. Peter Bernus, Ovidiu Noran, Arturo Molina. Enterprise Architecture: Twenty Years of the GERAM Framework, The International Federation of Automatic Control, Cape Town, South Africa. August 24-29, 2014
6. Guy Doumeingts, David Chen, Bruno Vallespir, P. Fénié, François Marcotte: GIM (GRAI Integrated Methodology) and its Evolutions - A Methodology to Design and Specify Advanced Manufacturing Systems. DIISM 1993: 101-120
7. Dennis E Wisnosky and Joseph Vogel. In Managing and Executing Projects to Build Enterprise Architectures Using the Department of Defense Architecture Framework (DoDAF). 2004.
8. Joseph Barjis. Enterprise, organization, modeling, simulation: putting pieces together. Proceeding of EOMAS, 2008.
9. Joseph Barjis and Alexander Verbraeck. The relevance of modeling and simulation in enterprise and organizational study. In Enterprise and Organizational Modeling and Simulation, pages 15-26. Springer, 2010.
10. F Vernadat., "UEML: towards a unified enterprise modelling language". In: Int. J. Production Research, 40 (17), 4309-4321. 2002
11. OMG Document, Business Process Model and Notation, http://www.omg.org/spec/BPMN/2.0/, formal/2011-01-03, January 2011.
12. A. van Lamsweerde, E. Letier. From Object Orientation to Goal Orientation: A Paradigm Shift for Requirements Engineering. Proc. Radical Innovations of Software and Systems Engineering, LNCS, 2003.
13. Eric Yu, Markus Strohmaier, and Xiaoxue Deng. Exploring intentional modeling and analysis for enterprise architecture. In Enterprise Distributed Object Computing Conference Workshops, 2006. EDOCW'06. 10th IEEE International, IEEE, 2006.

14. Object Management Group. Business Motivation Model v. 1.1, 2010. www.omg.org/spec/BMM/1.1/.
15. Donella H Meadows. Thinking in systems: A primer. Chelsea Green Publishing, 2008.
16. John Krogstie (2008) Using EEML for Combined Goal and Process Oriented Modeling: A Case Study. IDI, NTNU,Trondheim, Norway. Proceedings of EMMSAD 2008.
17. Iacob M, et al. State of the art in architecture support, ArchiMate deliverable D3.1. Enschede, The Netherlands: Telematica Instituut; 2003.
18. David Chen, Guy Doumeingts, François B. Vernadat: Architectures for enterprise integration and interoperability: Past, present and future. Computers in Industry 59(7): 647-659 (2008)
19. Fox, M.S., (1993), "Issues in Enterprise Modelling", Proceedings of the IEEE Conference on Systems, Man and Cybernetics , Le Toquet France, IEEE.
20. Parsons, Talcott, and I. Jones. *Structure and process in modern societies*. Vol. 3. New York: Free Press, 1960.
21. Balbir S. Barn, Tony Clark, Vinay Kulkarni: Next Generation Enterprise Modelling - The Role of Organizational Theory and Multi-Agent Systems. ICSOFT EA 2014: 482-487
22. Carl Hewitt. Actor model of computation: scalable robust information systems. arXiv preprint arXiv:1008.1459, 2010.
23. Carl Hewitt. Norms and commitment for iorgs (tm) information systems: Direct logic (tm) and participatory grounding checking. arXiv preprint arXiv:0906.2756, 2009.
24. Frank Van Harmelen, Vladimir Lifschitz, and Bruce Porter. Handbook of knowledge representation, volume 1. Elsevier, 2008.
25. Benjamin Camus, Christine Bourjot, Vincent Chevrier. Combining DEVS with Multi-agent Concepts to Design and Simulate Multi-models of Complex Systems. 2015.
26. Bollen, P.W.L. Using fact-based modeling for business simulation. Lecture Notes in Business Information processing 63, pp. 59-78, 2010, Springer-Verlag, Berlin